Sample Technical Due Diligence Report

Resource Planning Platform

Abridged public sample of a full technical assessment

PUBLIC SAMPLE

ASSESSMENT DATE

September 2026

REVIEW TIER

Source review with scanner support and selected AWS infrastructure evidence

This document is based on a real Technical Due Diligence assessment of a software product. Product, repository, and infrastructure identifiers have been anonymized.

Representative findings, severity ratings, analysis, and recommendations are preserved. Selected implementation details, supporting evidence, and sections have been shortened or omitted for the public version.

01 Assessment Scope

The assessed product is a resource planning platform. An Angular single-page application calls a Yii2 API. MySQL stores application data and report jobs. RabbitMQ carries change events to a separate notification service.

The review covered:

- Angular SPA
- Yii2 API
- Database migrations
- Report jobs
- Event publishers
- Automated tests
- Deployment files
- Selected AWS infrastructure evidence

A read-only AWS inspection covered selected records from:

- RDS
- EKS configuration
- ECR
- S3
- Parameter Store metadata
- Route53
- CloudWatch
- Cost Explorer

The review did **not** establish complete coverage of Kubernetes objects, application telemetry, or the notification consumer source. Where evidence was insufficient, the relevant area remained **Unassessed** rather than receiving an assumed low score.

WHAT THE FULL REPORT ADDS

The complete assessment includes repository-level evidence references, verification steps, dated infrastructure observations, evidence limitations, and remaining evidence requests.

02 Executive Summary

The main risks concern credentials, access control, and data loss.

The assessment identified authentication weaknesses that could allow credential exposure or continued token use beyond the intended lifetime. Enabled accounts also appeared to have broad administrative access, although that finding depends on the intended permission policy.

A destructive control-flow defect was identified in project deletion: a rejected delete operation could remove related planning records before returning an error.

The repositories also contain useful controls, including domain modules, API annotations, image checks, secret scans, manual deployment steps, and Sentry error reporting. However, application-test enforcement, health checks, migration recovery, and event delivery require correction.

The AWS inspection recorded an encrypted, private database with daily backups and seven-day retention. The database used a single availability zone. Recovery warnings, monitoring gaps, and missing deletion protection reduced confidence in resilience.

The assessment identified:

SEVERITY	COUNT	MEANING
HIGH	5	Credential, access, or data-loss exposure requiring prompt attention
MEDIUM	15	Reliability, data handling, or maintenance risk requiring planned correction
LOW	1	Limited issue requiring correction in the affected interface

One High finding depends on the intended permission policy.

Hosting and backup configuration were partly assessed. Successful restore, application capacity, and total ownership cost remained unproven.

03 Findings Summary

Severity describes impact and urgency. Confidence describes the supporting evidence. These are separate judgments.

A finding can have strong source evidence while its production impact remains unknown.

ID	FINDING	SEVERITY	MAIN IMPACT	RECOMMENDED ACTION
M01	Tracked files contain credentials	HIGH	Repository readers can receive credential values of unknown current validity	Rotate affected credentials and remove tracked values
M02	Login flow accepts an untrusted return destination	HIGH	A prepared authentication flow can disclose a user token	Bind login state to the session and restrict destinations
M03	Authentication does not enforce recorded token expiry	HIGH	A retained token can remain usable after its intended expiry	Reject expired tokens during authentication
M04	Enabled accounts share broad administrative access	HIGH policy-dependent	A compromised account can reach broad datasets	Define and enforce the permission policy
M05	A rejected project deletion removes related records	HIGH	An active project can lose assignments and contracts	Check eligibility before any deletion
M06	Request URLs contain reusable bearer tokens	MEDIUM	Systems retaining URLs can also retain credentials	Remove reusable tokens from URLs
M07	Report creation and job submission are not coordinated	MEDIUM	A report or its background job can be created without the other	Create both records transactionally
M08	Database changes and event delivery can disagree	MEDIUM	Users can miss updates after a successful save	Store pending events with the database change
M09	Installation and migration recovery need manual repair	MEDIUM	New environments and interrupted upgrades require intervention	Test clean installation and interrupted migrations
M10	Test and coverage checks leave delivery gaps	MEDIUM	Changes can merge without application-test enforcement	Run tests before merge and import coverage
M11	The SPA uses unsupported framework/runtime releases	MEDIUM	Maintenance and dependency updates become harder	Upgrade through a tested compatibility path
M12	API probes do not check application readiness	MEDIUM	A healthy proxy can conceal application or worker failure	Add application readiness and worker-progress checks
M13	Resource budgets are absent and optional autoscaling is defective	MEDIUM	Scheduling lacks declared budgets and scaling cannot work as configured	Set resource budgets and repair scaling configuration
M14	Restore workflows copy production data without masking	MEDIUM	Production records can reach lower environments	Use sanitized data and control exceptional copies
M15	Date navigation adds persistent subscriptions	MEDIUM	Long sessions can repeat work and retain state	Correct subscription lifetimes

ID	FINDING	SEVERITY	MAIN IMPACT	RECOMMENDED ACTION
M16	Reports collect broad upstream datasets before filtering	MEDIUM	Narrow selections can still require broad retrieval and storage	Filter upstream and process incrementally
M17	Deployment defaults leave container permissions broad	MEDIUM	Container compromise can expose unnecessary privileges	Restrict runtime privileges and token mounts
M18	Calendar controls lack basic keyboard support	LOW	Keyboard users can have difficulty operating affected controls	Use accessible controls and set document language
M19	AWS monitoring controls have gaps	MEDIUM	Failures and security events can lack configured detection	Check alerting, image scanning, and audit logs
M20	Database recovery warning and missing deletion protection	MEDIUM	Recovery assurance is weakened	Identify affected tables, enable protection, and test recovery
M21	Production credential records show old modification dates	MEDIUM	Available evidence does not establish recent credential replacement	Check ownership, revocation, and replacement history

PUBLIC SAMPLE NOTE

The full report expands each finding with supporting evidence, verification steps, limitations, and implementation-specific recommendations.

This public version expands **five representative findings** below.

04 Architecture Overview

The diagram below reflects the reviewed source configuration.

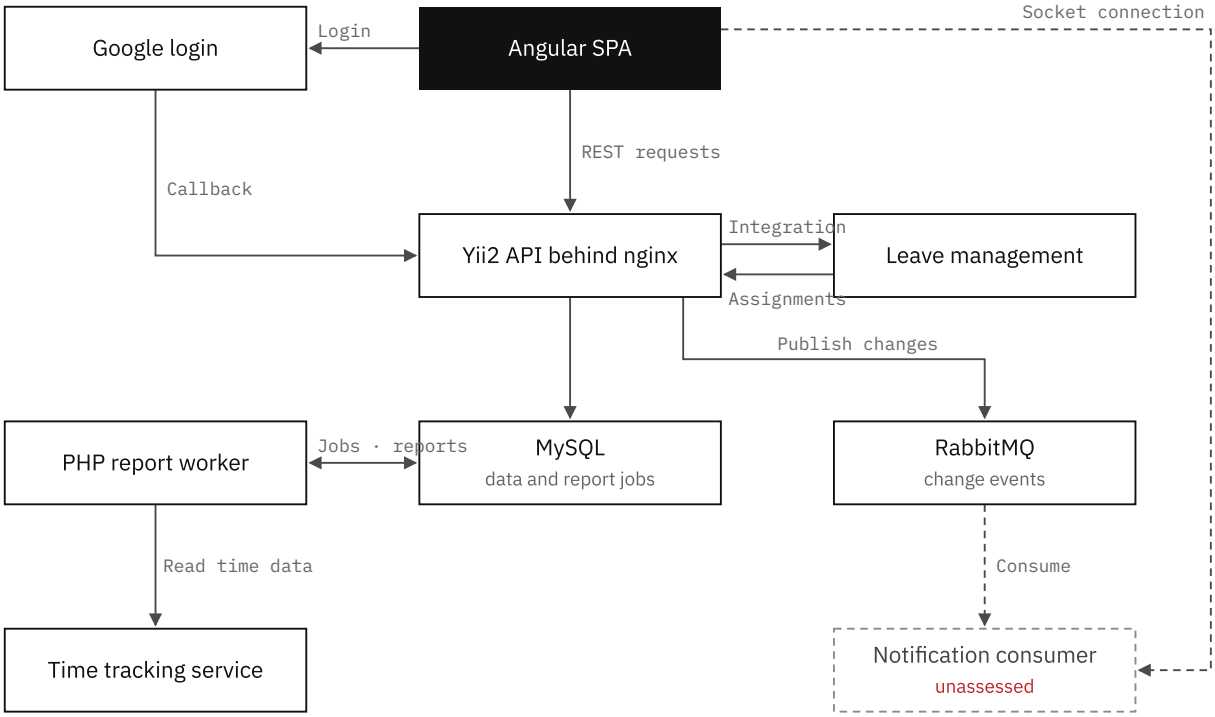

Figure 1. Architecture as reflected in the reviewed source configuration. Dashed paths were not fully assessed.

The API, report worker, and event publisher have different failure modes and therefore require separate assessment.

The notification consumer was visible as part of the system architecture, but its source was not available for complete review. The full report therefore treats conclusions about that path as limited rather than inferred.

05 Selected Detailed Findings

M02 HIGH

Login flow accepts an untrusted return destination

FINDING

The reviewed authentication flow accepted a return destination without sufficiently restricting where authentication output could be sent.

The behavior was reproduced using a synthetic token and a mocked identity service. No real account or external identity request was used during the check.

BUSINESS IMPACT

A prepared authentication flow could disclose a user token to an external destination.

RECOMMENDATION

- Bind random, expiring, single-use state to the initiating session
- Permit only approved return destinations
- Return a short-lived code for server-side exchange rather than a reusable bearer token
- Add tests covering rejected destinations and expired state

DETAILED EVIDENCE OMITTED FROM PUBLIC SAMPLE

The full report includes source references, the executed verification path, evidence limitations, and implementation-specific remediation notes.

M05 HIGH

A rejected project deletion removes related records

FINDING

The deletion flow removed related planning records before checking whether the project itself was eligible for deletion.

When the project was active, the operation then returned an error. Because the earlier changes were not restored, the rejected operation could still remove related data.

The control flow was reproduced using the actual controller method with isolated test relations. No production records were changed.

BUSINESS IMPACT

A user can receive a failed deletion response while related assignments and contracts have already been removed.

This creates a direct data-integrity risk because the system can report that an operation failed while still partially applying it.

RECOMMENDATION

- Check deletion eligibility before any mutation
- Apply related changes inside a database transaction
- Verify that the affected tables support transactional recovery
- Add an integration test proving that rejected deletion preserves every related record

---

DETAILED EVIDENCE OMITTED FROM PUBLIC SAMPLE

The full report includes the exact call order, source references, and related database-engine dependencies.

M08 MEDIUM

Database changes and event delivery can disagree

FINDING

Application changes were saved to the database and then published separately to the messaging layer.

The reviewed path did not include a stored retry record or equivalent mechanism guaranteeing that a successfully saved database change would eventually result in a corresponding event.

BUSINESS IMPACT

A database update can succeed while the related notification fails.

The application can therefore hold the correct state while downstream users or systems continue working from stale information.

RECOMMENDATION

- Store a pending event in the same transaction as the application change
- Publish pending events asynchronously
- Retry failed delivery
- Define duplicate-handling behavior before changing the delivery model

This pattern is commonly implemented as a transactional outbox.

DETAILED EVIDENCE OMITTED FROM PUBLIC SAMPLE

The complete report includes source-path analysis, messaging limitations, and the remaining consumer-side evidence required to validate delivery behavior end to end.

M12 MEDIUM

API probes do not check application readiness

FINDING

The inspected health probes confirmed that the proxy layer was responding, but did not prove that the application runtime or background worker was healthy.

A healthy proxy could therefore conceal an application or worker failure.

BUSINESS IMPACT

Infrastructure can appear healthy to the deployment platform while application functionality is unavailable or background work has stopped progressing.

This can delay failure detection and automated recovery.

RECOMMENDATION

- Add an application-level readiness check
- Monitor worker progress using a heartbeat or job-age metric
- Test isolated application and worker failures
- Avoid restarting otherwise healthy services solely because an external integration is unavailable

---

DETAILED EVIDENCE OMITTED FROM PUBLIC SAMPLE

The full report includes deployment-template evidence and monitoring dependencies.

M20 MEDIUM

Database recovery warning and missing deletion protection

FINDING

The infrastructure review identified a database recovery warning associated with some table types. Deletion protection was also disabled.

The review did not establish which tables were affected, so it did not assume that all application data shared the same recovery limitation.

BUSINESS IMPACT

Recovery confidence is reduced until the affected tables are identified and restore behavior is tested.

The absence of deletion protection also removes an additional safeguard against accidental database deletion.

RECOMMENDATION

- Identify affected tables and their business purpose
- Confirm the storage engine used by data involved in transactional repairs
- Enable deletion protection
- Test an isolated snapshot restore
- Test point-in-time recovery
- Check recovered records and measure recovery duration

DETAILED EVIDENCE OMITTED FROM PUBLIC SAMPLE

The complete assessment includes dated infrastructure observations, recovery evidence, and the limits of what could be verified without connecting directly to the database.

ADDITIONAL DETAILED FINDINGS IN THE FULL REPORT

The complete report also expands findings covering credentials, token expiry, authorization policy, report-job consistency, installation and migration recovery, test enforcement, dependency lifecycle, production-data handling, frontend subscription behavior, reporting performance, runtime permissions, accessibility, monitoring, and credential-rotation evidence.

06 Prioritized Recommendations

The order below reflects impact and dependencies.

The assessment does not assign repair cost or delivery dates without engineering input.

01 **Close credential and authentication defects.**

Rotate exposed credentials, confirm revocation, restrict authentication destinations, and enforce token expiry.

02 **Prevent destructive rejection.**

Move eligibility checks before mutations and prove that failed operations preserve related records.

03 **Establish the permission policy.**

Define server-side authorization boundaries and test cross-user access.

04 **Coordinate database writes and background work.**

Create dependent records transactionally and make event delivery recoverable.

05 **Add delivery checks before upgrades.**

Run application tests before merge, improve coverage visibility, test clean installation and migration recovery, then upgrade unsupported dependencies.

06 **Correct application checks and deployment defaults.**

Add application readiness, monitor worker progress, set resource budgets, repair optional scaling, and restrict runtime permissions.

07 **Control production-data copies.**

Use sanitized datasets and define approval, access, and retention rules for exceptional copies.

08 **Bound browser and report work.**

Correct subscription lifetimes, filter upstream data, and measure representative long-running workloads.

09 **Correct verified accessibility defects.**

Restore keyboard access and review remaining static-analysis alerts in context.

10 **Correct AWS control gaps.**

Check alerting and image-scanning coverage, enable relevant audit logs and deletion protection, investigate recovery warnings, and test recovery.

11 **Obtain remaining operating evidence.**

Review live workload configuration, external alert rules, incident records, complete cost allocation, representative workload measurements, and currently unassessed components.

07 Evaluation Scores

Scores apply only to criteria supported by reviewed evidence.

Missing access does not receive a score of 1.
Where evidence is insufficient, the result is **Unassessed**.

SCORE	MEANING
1	Major demonstrated deficiencies affect the criterion
2	Relevant controls exist, with material deficiencies
3	Reviewed evidence shows an acceptable control design; operating effectiveness can remain unassessed
4	Evidence shows consistent practice with minor gaps
5	Evidence shows the criterion is fully met, including operating effectiveness
UNASSESSED	Evidence does not support a defensible score
N/A	The criterion does not apply to the assessed scope

SELECTED EVALUATION AREAS

INFRASTRUCTURE

EVALUATION CRITERIA	SCORE	EVIDENCE AND LIMITS
Hosting and cloud services	3	Reviewed AWS evidence showed private, encrypted database hosting and shared container infrastructure. Some live workload configuration remained unavailable.
Scalability	2	Resource budgets and optional autoscaling required correction. Capacity was not measured.
Fault tolerance and high availability	2	The database used one availability zone. Application replica behavior and tested failover remained unverified.
Cost of ownership	UNASSESSED	Tagged database cost was visible, but shared infrastructure and support effort were not fully allocated.

SECURITY

EVALUATION CRITERIA	SCORE	EVIDENCE AND LIMITS
Access control	1	Authentication defects were demonstrated. Intended authorization boundaries required confirmation.
Data security	1	Credential handling and token exposure required correction. Encryption controls were present in reviewed AWS evidence.
Network security	2	The database was private and infrastructure access had restrictions, but effective reachability and workload-level policies were not fully verified.
Compliance	UNASSESSED	Applicable obligations and assurance evidence were not available.

APPLICATION ARCHITECTURE

EVALUATION CRITERIA	SCORE	EVIDENCE AND LIMITS
Code quality and maintainability	2	Modules separated several concerns, but specific mutation and subscription defects increased change risk.
Error handling	2	Logging and status controls existed, while some failure paths could leave inconsistent results.
API and integration	2	Documented endpoints and separate jobs existed. Event recovery and upstream request handling required correction.

DEPLOYMENT AND CI/CD

EVALUATION CRITERIA	SCORE	EVIDENCE AND LIMITS
Build and deployment pipelines	3	Source defined image checks and staged promotion. Consistent operating effectiveness was not independently proven.
Automation	2	Some automated tests existed, but merge-time enforcement and coverage checks had gaps.
Rollback strategies	2	Application rollback mechanisms existed. Database recovery remained unproven.
Dependency updates	2	Lockfiles existed, but the SPA used unsupported releases.

DISASTER RECOVERY

EVALUATION CRITERIA	SCORE	EVIDENCE AND LIMITS
DR plan	UNASSESSED	Backup mechanisms were visible. A written recovery plan and successful restore record were not available.
RPO and RTO	UNASSESSED	Recovery point and recovery time objectives were not supplied.

ADDITIONAL EVALUATION AREAS IN THE FULL REPORT

Performance, Database Management, Monitoring & Logging, Compliance & Documentation, User Experience, Innovation & Future Roadmap, and Extendability are assessed separately in the complete report.

08 Risk Assessment

The table below groups the individual findings into business risk areas without counting the same defect twice.

RISK AREA	FINDING IDS	SEVERITY	BUSINESS CONSEQUENCE	REQUIRED EVIDENCE AFTER CORRECTION
Credentials and access	M01–M04, M06	HIGH with one policy-dependent finding	A disclosed credential or compromised account can grant broad access	Check rotation, login destinations, expiry, and permission denials
Data integrity	M05, M07	HIGH	Rejected or incomplete operations can remove records or create unusable reports	Test rejected deletions and transactional failure paths
Delivery and recovery	M08–M10	MEDIUM	Events, migrations, or untested changes can leave inconsistent application state	Check event retries, migration recovery, and mandatory tests
Runtime reliability	M11–M13, M15–M17	MEDIUM	Unsupported dependencies, broad permissions, and weak failure detection increase operational exposure	Check updated images, runtime controls, probes, and representative workloads
Production-data handling	M14	MEDIUM	Unmasked copies can extend access beyond production	Check masking, destination access, copy history, and retention
Interface access	M18	LOW	Keyboard users can have difficulty operating affected controls	Test affected interactions with a keyboard
AWS detection and credentials	M19, M21	MEDIUM	Missing monitoring controls and old credential records can delay detection or replacement	Check alert delivery, scan results, audit logs, and revocation
Database recovery	M20	MEDIUM	Recovery warnings and missing deletion protection weaken recovery assurance	Identify affected tables and test snapshot and point-in-time restores
Remaining operations	Unassessed	UNASSESSED	Backup configuration was visible, while recovery success, capacity, and total cost remained unproven	Obtain remaining records and run representative recovery and workload tests

09 Evidence and Limits

A Technical Due Diligence conclusion is only as strong as the evidence behind it.

The full assessment separates:

- confirmed source observations
- reproduced behaviors
- dated infrastructure observations
- scanner alerts requiring manual review
- assumptions
- unassessed areas

EXAMPLES OF CHECKS PERFORMED

CHECK	RESULT	LIMIT
Source inspection	Traced authentication, permissions, deletion, reports, events, subscriptions, and deployment configuration	Source inspection does not prove production behavior in every case
Authentication return check	Reproduced token return behavior using a synthetic token and mocked identity service	No real identity account or production credential was used
Project-deletion check	Reproduced the destructive control flow using isolated test relations	No production database rows were changed
AWS inspection	Recorded selected configuration, backup events, database metrics, and cost data	Live Kubernetes objects, application telemetry, and full shared-cost allocation were unavailable

ANALYSIS COVERAGE

METRIC	API	SPA
Analyzed lines of code	17,419	24,287
Duplicated lines	9.2%	3.9%
Test coverage	Unmeasured	Unmeasured

These metrics describe the analyzed file set. They do not, by themselves, establish maintainability or application quality.

Static-analysis alerts were reviewed as leads rather than treated automatically as confirmed business risks.

FULL EVIDENCE APPENDIX OMITTED

The complete report includes source snapshots, exact evidence references, performed checks and outcomes, AWS observation scope, analysis coverage, remaining evidence requests, and explicit limits on each conclusion.

10 What Is Not Shown in This Public Sample

This sample is intentionally shorter than the original assessment.

The full report contains:

- all 21 findings expanded individually
- source-level evidence references
- verification steps and test outcomes
- implementation-specific remediation notes
- complete evaluation tables
- detailed infrastructure observations
- scalability constraints and uncertainties
- backup and disaster-recovery analysis
- monitoring and logging analysis
- source snapshot and review-scope records
- remaining evidence requests
- explicit limitations on conclusions

The goal of the public sample is to show the **depth, structure, and decision value** of the deliverable without publishing every implementation detail.

Need Technical Due Diligence?

If you're evaluating a software product, preparing for a transaction, or have questions about the assessment process, email us at **solutions@mev.com**.

We can discuss your scope, answer questions about the report, or **share the full sample assessment**.

---

solutions@mev.com